United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Final Exam : Fall 2022

CSE 4889: Machine Learning: Section: A

Total Marks: **40** Duration: 2.00 hrs

There are 10 questions. **Answer 8 questions**. Every question carries equal marks

****Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.**

1. You know that XOR function cannot be implemented by a single layer neural network. Show in a step by step fashion that XOR function can be implemented by a multilayer neural network

2. Assume that you have a two input neural network. You have three hidden neurons in the first hidden layer, one output neuron in the output layer and you are using a step function as the activation function. Show geometrically that the input data is mapped into vertices of a three dimensional unit length cube by the hidden layer neurons. What is the function of the output neuron?

3. What steps can you take to avoid local minima during training of a neural network? How does momentum term increase the convergence speed of training a neural network?

4. What is cross entropy function? When is it minimum and when is it maximum? What has its advantage over other cost functions? What should be the activation function of the output nodes if you use cross entropy cost function and why?

5. How can you control the size of a neural network using parameter sensitivity calculation and cost function regularization

6. From the definition of support vector machine formulate the Lagrangian optimization problem. Now derive the following where the symbols carry their usual meaning:

$$w = \sum_{i=1}^{N} \lambda_i y_i x_i$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0$$

Which examples are the support vectors? What are the values of Lamdas for support vectors? For which training examples the Lamda values are zeros?

7. Three men are tossing coins with the probabilities of coming up head and tail are <0.4,0.6>, <0.7,0.3> and <0.5,0.5> respectively. If the tossing is done by the 1st man, the probability of tossing next by the 1st, 2nd and 3rd man are 0.4, 0.3 and 0.3 respectively. If the tossing is done by the 2nd man, the probability of tossing next by the 1st, 2nd and 3rd man are 0.5, 0.2 and 0.3 respectively. If the tossing is done by the 3rd man, the probability of tossing next by the 1st, 2nd and 3rd man are 0.4, 0.2 and 0.4 respectively. The three men tossed four times and the outcome was H, T, H, H. Assume that the probability of tossing will be started by 1st, 2nd or 3rd man are 0.4, 0.3 and 0.3 respectively. Find the optimal state sequence to make the probability of the given outcome maximum.

8. An input image 227X227X3 is convolved with 100 filters having size = 21X21X3, stride = 4 and zero padding = 1. In the 2nd phase output is convolved with 50 filters having size = 7X7X100, stride = 2 and zero padding = 0. In the third phase the output is convolved with 10 filters having size = 5X5X50, stride = 1 and zero padding = 0. Find out the sizes of the outputs after each phase. What will be the total number of neurons needed?

9. In Question #8, show the implementation of the 1st phase using matrix multiplication. If you want to avoid pooling for downsizing the output, what alternative can you use?

10. How can you calculate the probability of a sequence of observation under Hidden Markov Model using forward procedure? What is its complexity?

CSE 4889 (Machine Learning)

Class Test 3

Section B

Time 20 minutes

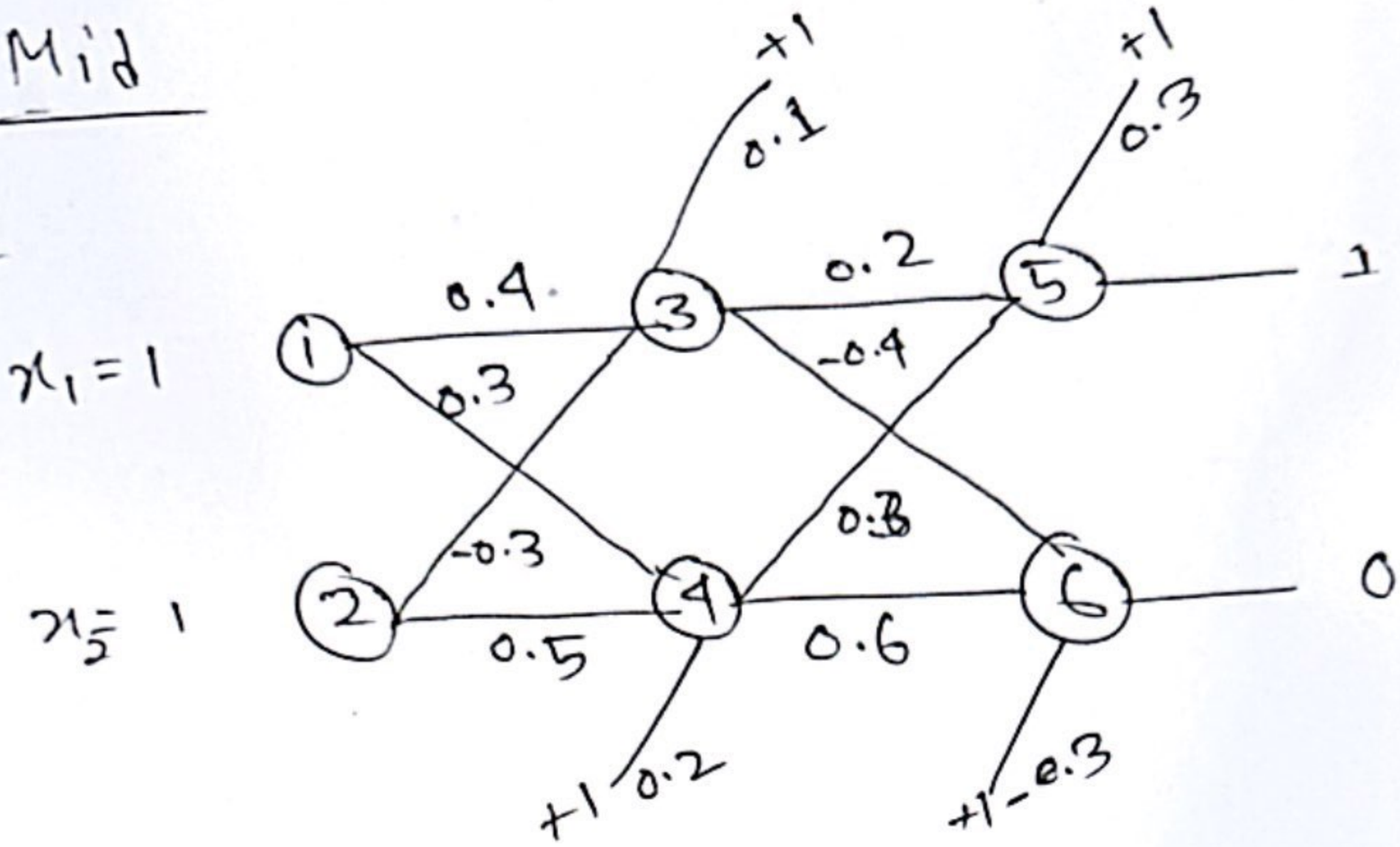
Student Id: 01120120 5

1. In a two input neural network there are three neurons in the first hidden layer and one neuron in the output layer. Show the polyhedral regions formed by the hidden units and mark the regions which map into 110 and 111. What is the dimension of the plane produced by the output unit?

2. *There are two output units in a multilayer neural network. For some input example the outputs are 0.8 and 0.2 respectively where the targets are 1.0 and 0.0 respectively. For a 2nd example the outputs are 0.6 and 0.4 respectively where the targets are 0.0 and 1.0 respectively. Find the value of the cross entropy cost function for these two examples.*

Mid

81



$I_j \quad \Sigma(input_j \times w_j)$

$I_3 = x_1 \times w_{13} + x_2 \times w_{23} + 1 \times w_{03}$

$= 1 \times 0.4 + 1 \times (-0.3) + 1 \times 0.1$

$= 0.2$

$I_4 = x_1 \times w_{14} + x_2 \times w_{24} + 1 \times w_{04}$

$= 1 \times 0.3 + 1 \times 0.5 + 1 \times 0.2$

$= 1$

$I_5 = O_3 \times w_{35} + O_4 \times w_{45} + 1 \times w_{05}$

$= 0.55 \times 0.2 + 0.73 \times 0.3 + 1 \times 0.3$

$= 0.629$

$I_6 = O_3 \times w_{36} + O_4 \times w_{46} + 1 \times w_{06}$

$= 0.55 \times (-0.4) + 0.73 \times 0.6 + 1 \times (-0.3)$

$= -0.082$

$O_j \quad \frac{1}{1+e^{-I_j}}$

$O_3 = \frac{1}{1+e^{-0.2}}$

$= 0.55$

$O_4 = \frac{1}{1+e^{-1}}$

$= 0.73$

$O_5 = \frac{1}{1+e^{-0.629}}$

$= 0.65$

$O_6 = \frac{1}{1+e^{-(-0.082)}}$

$= 0.48$

$$\Delta_j = Error_j \times O_j \times (1 - O_j)$$

$$\Delta_5 = (target - O_j) \times O_j \times (1 - O_j)$$

$$= (1 - O_5) \times O_5 \times (1 - O_5)$$

$$= (1 - 0.65) \times 0.65 \times (1 - 0.65)$$

$$= 0.08$$

$$\Delta_6 = (0 - 0.48) \times 0.48 \times (1 - 0.48)$$

$$= -0.12$$

$$\Delta_3 = Error_3 \times O_3 \times (1 - O_3)$$

$$= (\Delta_5 \times W_{35} + \Delta_6 \times W_{36}) \times O_3 \times (1 - O_3)$$

$$= \{0.08 \times 0.2 + (-0.12) \times (-0.4)\} \times 0.55 \times (1 - 0.55)$$

$$= 0.016$$

$$\Delta_4 = Error_4 \times O_4 \times (1 - O_4)$$

$$= (\Delta_5 \times W_{45} + \Delta_6 \times W_{46}) \times O_4 \times (1 - O_4)$$

$$= \{0.08 \times 0.3 + (-0.12) \times 0.6\} \times 0.73 \times (1 - 0.73)$$

$$= -0.0095$$

change the weight of every links

$$W_{ij} = W_{ij}(old) + \alpha \times O_i \times \Delta_6$$

$$W_{13} = W_{13}(old) + \alpha \times O_1 \times \Delta_3$$

$$= 0.4 + 0.9 \times 1 \times 0.016$$

$$= 0.414$$

$$W_{14} = W_{14}(old) + \alpha \times O_1 \times \Delta_4$$

$$= 0.3 + 0.9 \times 1 \times (-0.0095)$$

$$= 0.299$$

$$W_{23} = -0.3 + 0.9 \times 1 \times 0.016$$

$$= -0.29$$

$$W_{24} = 0.5 + 0.9 \times 1 \times (0.0095)$$

$$= 0.99$$

$$W_{35} = 0.2 + 0.9 \times 0.55 \times (0.08)$$

$$= 0.24$$

$$W_{36} = -0.4 + 0.9 \times 0.55 \times (-0.12)$$

$$= -0.46$$

$$w_{45} = 0.3 + 0.9 \times 0.73 \times 0.02$$

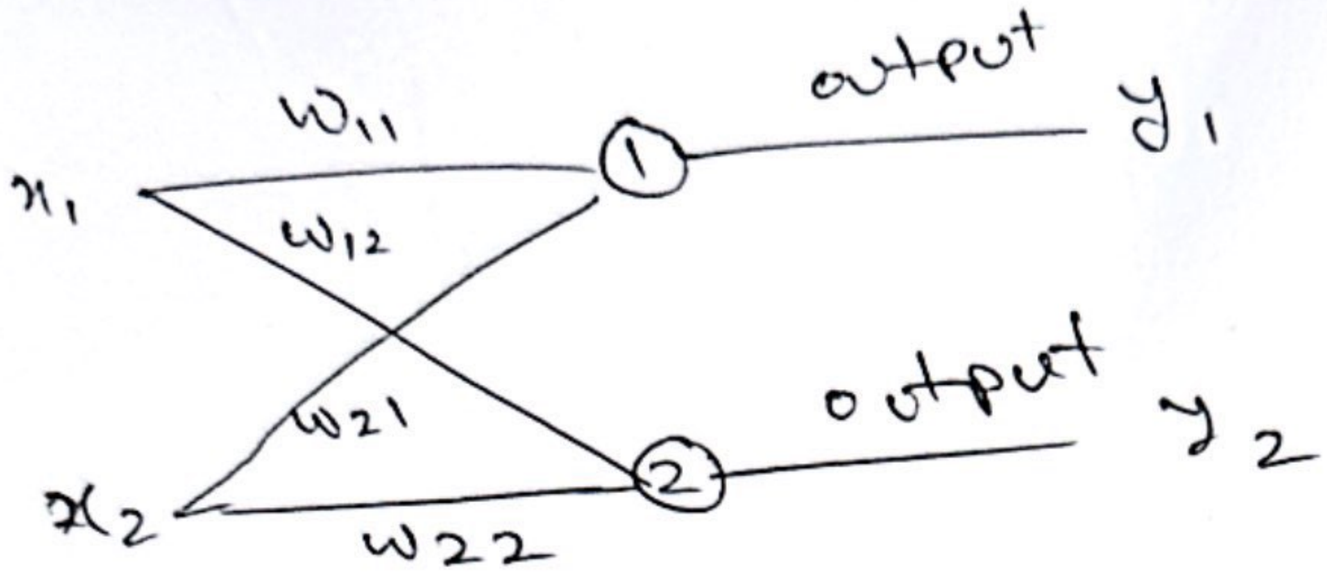
$$= 0.35$$

$$w_{46} = 0.6 + 0.9 \times 0.73 \times (-0.12)$$

$$= 0.52$$

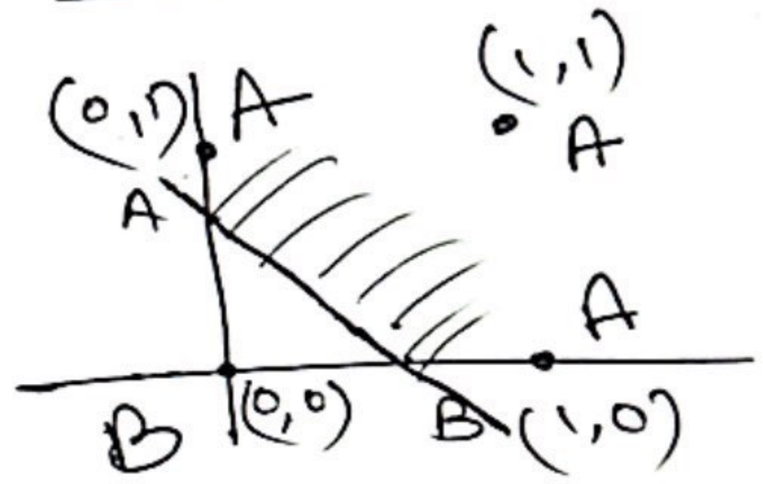
11

XOR function by multilayer neural network:



x_1	x_2
0	0
0	1
1	0
1	1

Let for y_1 output

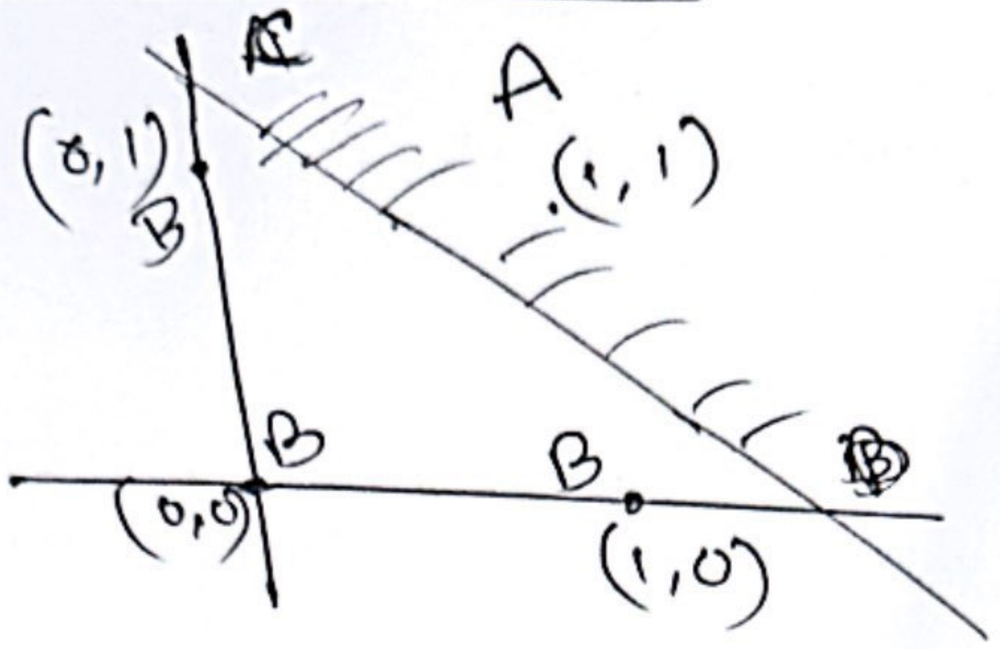


if for y, threshold becomes AB line then the output will be 0, 1, 1, 1

x_1	x_2	y_1
0	0	0
0	1	1
1	0	1
1	1	1

Link OR gate

for y_2 output

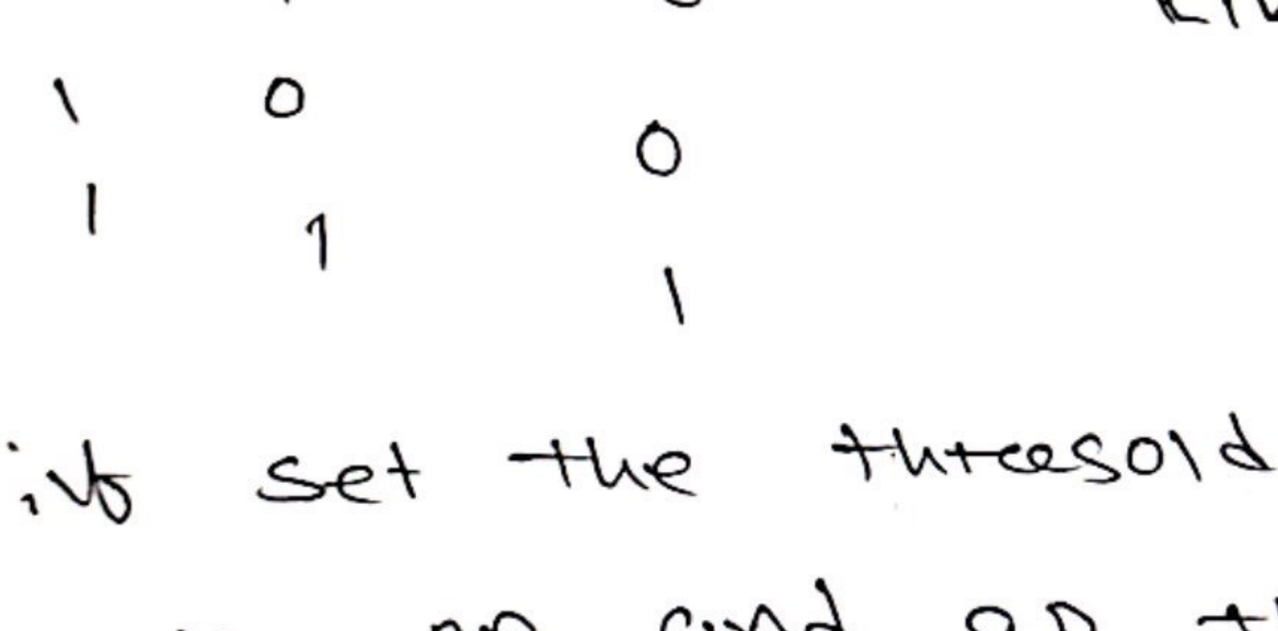


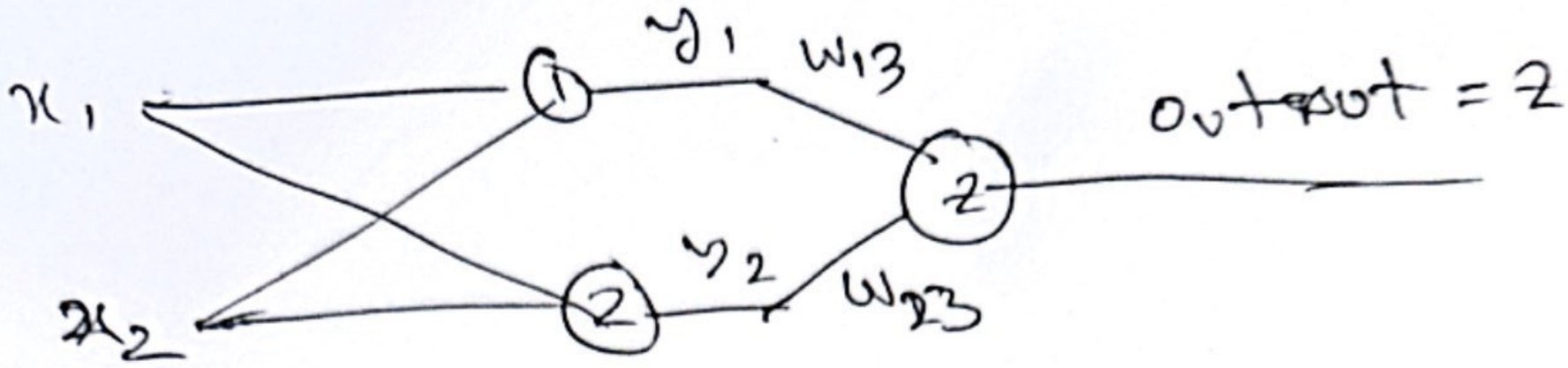
∴ if for y_2 threesold becomes CB line then the output will be 0, 0, 0, 1

x_1	x_2	y_2
0	0	0
0	1	0
1	0	0
1	1	1

Like AND gate

Now, if set the threesold a rang between AB and CD then the output will be change





y_1	y_2	z
0	0	0
1	0	1
1	0	1
1	1	0

So the z function act link XOR function ~~and~~ & when y_1 and y_2 are the input of z.

This is the way XOR function implimented by multilayer neural network.

CT

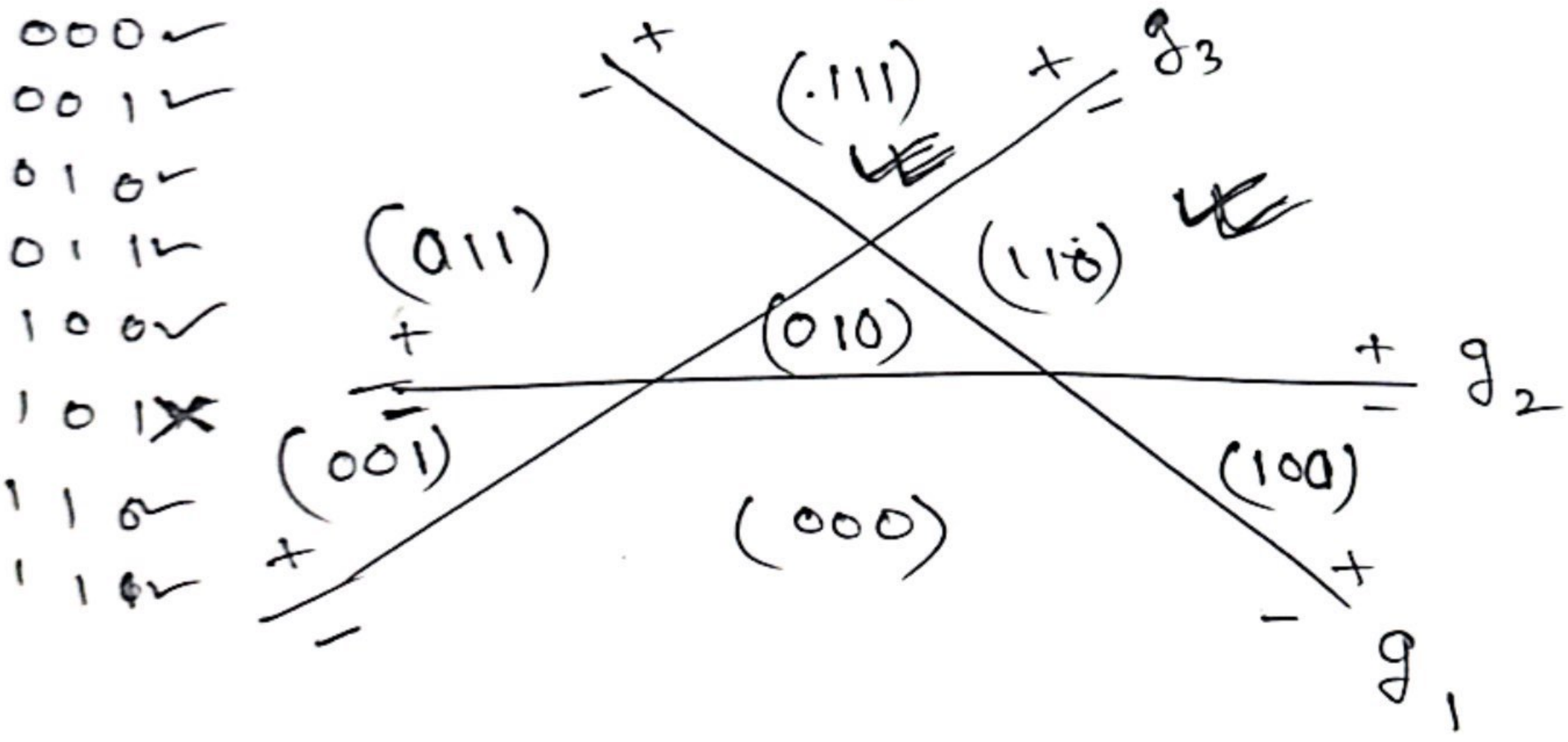
1) Given -

three neurons in the first hidden layer. And one output neuron

Polyhydral regions :

g_1 ; g_2 , g_3 as 3 neuron

(+) and (-) determines the input 1 and 0 ~~or~~ respectively.



~~what is +~~

~~output~~

[Note: 101 vertex does not corresponded to any regions of polyhydral. It means, it is ~~dimension of output unit~~ :

~~an~~ a viretual polyhydra. This will not influence the classification task.

As Hidden layer dimension is 3 (three input)

So, output layer plane dimension is : 2

31 To avoid local minima.

1) ~~Startin~~
1) Random Restart (Hill climbing):
- it can be started from a random position and update the weight and fix the error. The process to be continued.

2) Batch mode: After itiratting all the example and sumation of the error it should update the weight and fix it.

3) Pattern mode: After itirating every single example it fixes the error and update the weight.

For momentum term increase

$$\Delta W_{jk}(new) = \alpha \Delta W_{jk}(old) + \eta a_j \Delta_k$$

$$W_{jk}(new) = W_{jk}(old) + \Delta W_{jk}(new)$$

$$\therefore \Delta W_{jk}(t) = \alpha \Delta W_{jk}(t-1) + \eta g(t)$$

$$\Delta W_{jk}(1) = \alpha \Delta W_{jk}(0) + \eta g(1)$$

$$\Delta W_{jk}(2) = \alpha \Delta W_{jk}(1) + \eta g(2)$$

$$= \alpha \left(\alpha \Delta W_{jk}(0) + \eta g(1) \right) + \eta g(2)$$

$$= \alpha^2 \Delta W_{jk}(0) + \alpha \eta g(1) + \eta g(2)$$

$$\Delta W_{jk}(3) = \alpha \Delta W_{jk}(2) + \eta g(3)$$

$$= \alpha \left(\alpha^2 \Delta W_{jk}(0) + \alpha \eta g(1) + \eta g(2) \right) + \eta g(3)$$

$$= \alpha^3 \Delta W_{jk}(0) + \alpha^2 \eta g(1) + \alpha \eta g(2) + \eta g(3)$$

$$= \alpha^3 \Delta W_{jk}(0) + \eta \left(\alpha^2 g(1) + \alpha^1 \eta(2) + \alpha^0 g(3) \right)$$

$$\Delta w_{jk}(T) = \alpha^T \Delta w_{jk}(0) +$$

$$\eta\left(\alpha^{T-1} g(1) + \alpha^{T-2} g(2) + \cdots - g(T)\right)$$

$$\Delta w_{jk}(T) = \alpha^T \Delta w_{jk}(0) + \eta \sum_{t=0}^{T-1} \alpha^t g(T-t)$$

$$= \eta\left(1 + \alpha + \alpha^2 + \cdots + \alpha^{T-1}\right) g$$

$$= \eta \frac{1}{1-\alpha} \times g$$

$$= \frac{\eta}{1-\alpha} \times g \quad \left[\text{where } g \text{ is gradient which is very small}\right]$$

So, momentum could be increase by increasing the value of η.

As $\Delta w_{jk}(T) \propto \eta$

And η is the controller of the contribution of the momentum term to the update direction.

4)

Cross entropy function:

- The cross-entropy function is a commonly used cost function in machine learning particularly in classification tasks. It measures the difference between the predicted probability distribution and the true probability distribution at the target variable.

- And it also a well formed function which doesn't trap at local minima.

$$T = \sum_{i=1}^{N} \sum_{k=1}^{k_0} \left(y_k^{(i)} \ln \hat{y}_k^{(i)} + \left(1 - y_k^{(i)}\right) \ln\left(1 - \hat{y}_k^{(i)}\right) \right)$$

T is the cross entropy cost function

where $y_k^{(i)}$ is actual output

$\hat{y}_k^{(i)}$ is predicted output

minimum T if $y_k(i) = \hat{y}_k(i)$

maximum T if $y_k(i) \neq \hat{y}_k(i)$ that means completely different

- The activation function could be sigmoid function for the output node.

Because: Sigmoid function maps the output of the last layer of the neural network to a value between 0 and 1.

CT- Section (B)

2) 1st example:

output : 0.8 , 0.2 ,

target : 1.0 , 0.0

$\therefore y_k(i) =$ 1, 0 (actual)

$\hat{y_k} =$ 0.8, 0.2 (prediction)

Cross-entropy cost function

$$L = \sum \sum \left[y_k(i) \ln \hat{y_k}(i) + (1 - y_k(i)) \ln (1 - \hat{y_k(i)}) \right]$$

$$= \sum \sum y \ln y' + (1-y) \ln (1-y')$$

$$= \left[y_1 \ln y'_1 + (1-y_1) \ln (1-y'_1) \right]$$

$$+ \left[y_2 \ln y'_2 + (1-y_2) \ln (1-y'_2) \right]$$

$$= \left[0.8 \ln (1) + (1-0.8) \ln (1-1) \right]$$

$$+ \left[0.2 \ln (0) + (1-0.2) \ln (1-0) \right]$$

$$= \left[1 \ln (0.8) + (1-1) \ln (1-0.8) \right]$$

$$+ \left[0 \times \ln (0.2) + (1-0) \ln (1-0.2) \right]$$

$$= \left[-0.223 + 0 \right] + \left[0 + (-0.223) \right]$$

$$= -0.446$$

for 2nd example

output: 0.6 , 0.4

target: 0 , 1

$$L = [0.6 \ln(0) + (1-0.6)\ln(1-0)] + [0.4 \ln(1) + (1-0.4)\ln(1-1)]$$

prediction, $y'_k = 0.6 , 0.4$

Actual, $y = 0 , 1$

$$L = [0 \times \ln(0.6) + (1-0)\ln(1-0.6)] + [1 \ln(0.4) + (1-1)\ln(1-0.4)]$$

$$= [0 + (-0.92)] + [1 \times (-0.92) + 0]$$

$$= -1.84$$

5] control
^ Size of Neural Network:

Cost function Regularization

$$J = \sum_{i=1}^{N} \mathcal{E}(i) + \alpha \mathcal{E}_p(\underline{w})$$

Here the size minimize by J and it depends on $\mathcal{E}_p(\underline{w})$ which is regularization factor.

$$\mathcal{E}_p(\underline{w}) = \sum_{k=1}^{k} b(w_k)^2$$

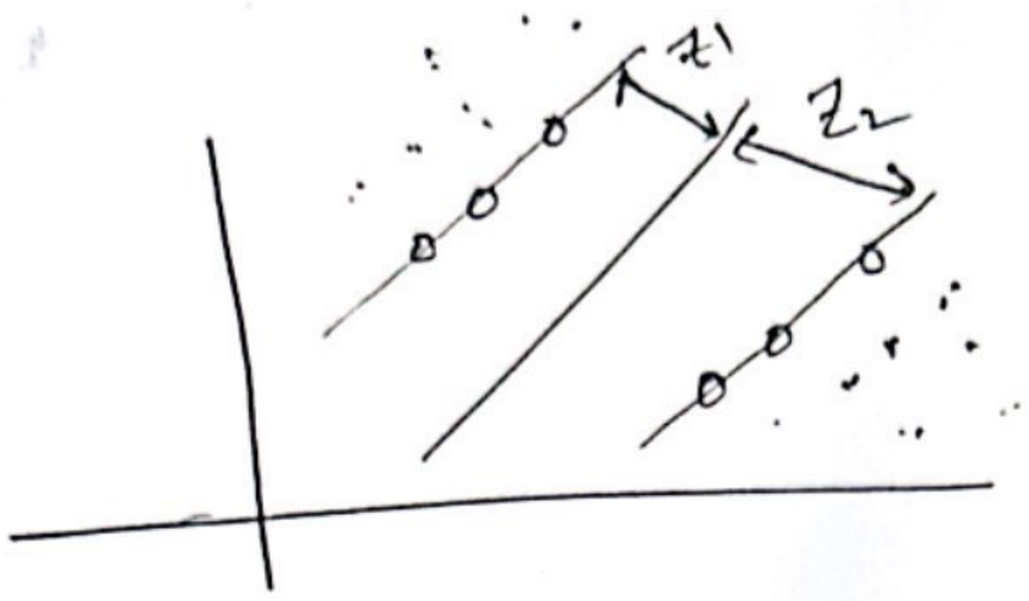
$$= \sum_{k=1}^{k} \frac{w_k^2}{w_0^2 + w_k^2}$$

if $\frac{w_k^2}{w_0^2 + w_k^2} <$ threshold

then this weighted example

ddete from ~~set~~ network.

61



(i) $\underline{w}^T \underline{x} + w_0 \geq 1$, $\forall x \in w_1$ [lie on the z1]

(ii) $\underline{w}^T \underline{x} + w_0 \leq -1$, $\forall x \in w_2$ [z_2]

from (i) $= \frac{1}{\|w\|}$ [अंतर]

from (ii) $= \frac{1}{\|w\|}$

merge these equations:

$$\frac{1}{\|w\|} + \frac{1}{\|w\|} = \frac{2}{\|w\|}$$

After minimizing, for classify 2 class by 1 equation :

$$f(w_1, w_2) = \frac{1}{2} \|w\|^2$$

subject to $y_i (w^T x_i + w_0) \geq 1$

$i = 1, 2, 3 \ldots N$

And

$$\frac{dL}{dw}(w, w_0, \lambda) = 0 \longrightarrow (i)$$

$$\frac{dL}{dw_0}(w, w_0, \lambda) = 0$$

$$\lambda \geq 0 \longrightarrow (iii)$$

So, from

$$\Rightarrow y_i(w^T x_i + w_0) - 1 \geq 0 \longrightarrow (iv)$$

from (iii) and (iv)

$$\lambda_i \left[y_i (w^T_i + w_0) - 1 \right] \geq 0 \longrightarrow (v)$$

where, λ = Lagrangian multiplier

and $L(w, w_0, \lambda)$ is the Lagrangian function

$$L(w, w_0, \lambda) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \lambda_i y_i x_i$$

$$L(w, w_0, \lambda) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \lambda_i \left[y_i (w^T x_i + w_0) - 1 \right]$$

$$\frac{d}{dw} L(w, w_0, \lambda) = \frac{d}{dw} \left(\frac{1}{2} \times w^T w - \sum_{i=1}^{N} \left(\lambda_i y_i w^T x_i + \lambda_i y_i w_0 - \lambda_i \right) \right]$$

$$\Rightarrow 0 = \frac{1}{2}x2w - \sum_{i=1}^{N} \lambda_i y_i x_i + 0 - 0$$

$$\Rightarrow 0 = w - \sum_{i=1}^{N} \lambda_i y_i x_i$$

$$\therefore \boxed{w = \sum_{i=1}^{N} \lambda_i y_i x_i}$$

Again

$$\frac{d}{dw_0} L(w, w_0, \lambda) = \frac{d}{dw_0}\left[\frac{1}{2} w^T w - \sum_{i=1}^{N} \left(\lambda_i y_i w^T x_i + \lambda_i y_i w_0 - \lambda_i\right)\right]$$

$$\Rightarrow 0 = 0 - \sum_{i=1}^{N} (0 + \lambda_i y_i - 0)$$

$$\Rightarrow 0 = -\sum_{i=1}^{N} \lambda_i y_i$$

$$\therefore \boxed{\sum_{i=1}^{N} \lambda_i y_i = 0}$$

- which examples are the support vectors?

Ans: the examples that lie on the margin boundary or inside the margin boundary, whic means they have non-zero Lagragian multiplier (α)

$\therefore \alpha \neq 0$

- for support vector $\rightarrow$ $\alpha \neq 0$

- for non-support vectors means those that lie far away from the margin boundary, $\alpha = 0$

Dual Problem

$$L(w, w_0, \lambda) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \lambda_i \left[y_i (\underline{w}^T \underline{x}_i + w_0) - 1 \right] \longrightarrow (i)$$

As, $w = \sum_{i=1}^{N} \lambda_i y_i \underline{x}_i$

and $\sum_{i=0}^{N} \lambda_i y_i = 0$

So,

$$L(w, w_0 \lambda) = \frac{1}{2} \left(\sum_{i=1}^{N} \lambda_i y_i \underline{x}_i \right)^T \left(\sum_{i=1}^{N} \lambda_j y_j \underline{x}_j \right) - \left(\sum_{i=1}^{N} \lambda_i y_i \underline{x}_i^T \sum_{i=1}^{N} \lambda_i y_i + w_0 \sum \lambda_i y_i - \sum \lambda_i \right)$$

$$= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\lambda_i \lambda_j y_i y_j \underline{x}_i^T \underline{x}_j \right) - \left(\sum_{i=1}^{N} \left(\sum_{j=1}^{N} \lambda_j y_j \underline{x}_j \right)^T \lambda_i y_i \underline{x}_i + w_0 \times 0 - \sum \lambda_i \right)$$

$$= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\lambda_i \lambda_j y_i y_j \underline{x}_i^T \underline{x}_j \right) - \left(\sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i y_i y_j \underline{x}_i \underline{x}_j^T \right) + \sum \lambda_i$$

$$\Theta = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j \underline{x}_i^T \underline{x}_j + \sum_{i=1}^{N} \lambda_i$$

$$\therefore = \sum \lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j \underline{x}_i^T \underline{x}_j$$

Maximize equation

$\underline{x}_i^T \underline{x}_j$ dot product of two matrix.

71

Given:

$$B = \begin{bmatrix} & H & T \\ & 0.4 & 0.6 \\ & 0.7 & 0.3 \\ & 0.5 & 0.5 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.5 & 0.2 & 0.3 \\ 0.4 & 0.2 & 0.4 \end{bmatrix}$$

$$\pi = \begin{bmatrix} 0.4 \\ 0.3 \\ 0.3 \end{bmatrix}$$

H T H H

Formula: $-\ln(\pi_i b_i(o_1))$

And $-\ln(a_{ij}\, b_j(j))$

1st output : H

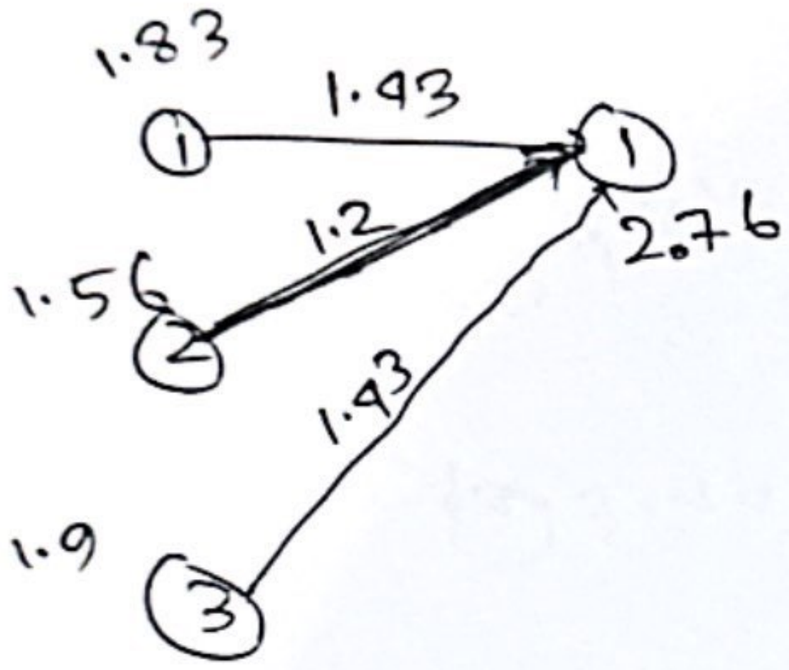
H

① → $-\ln(0.9 \times 0.9) = 1.83$

② → $-\ln(0.3 \times 0.7) = 1.56$

③ → $-\ln(0.3 \times 0.5) = 1.9$

For 2nd output : T



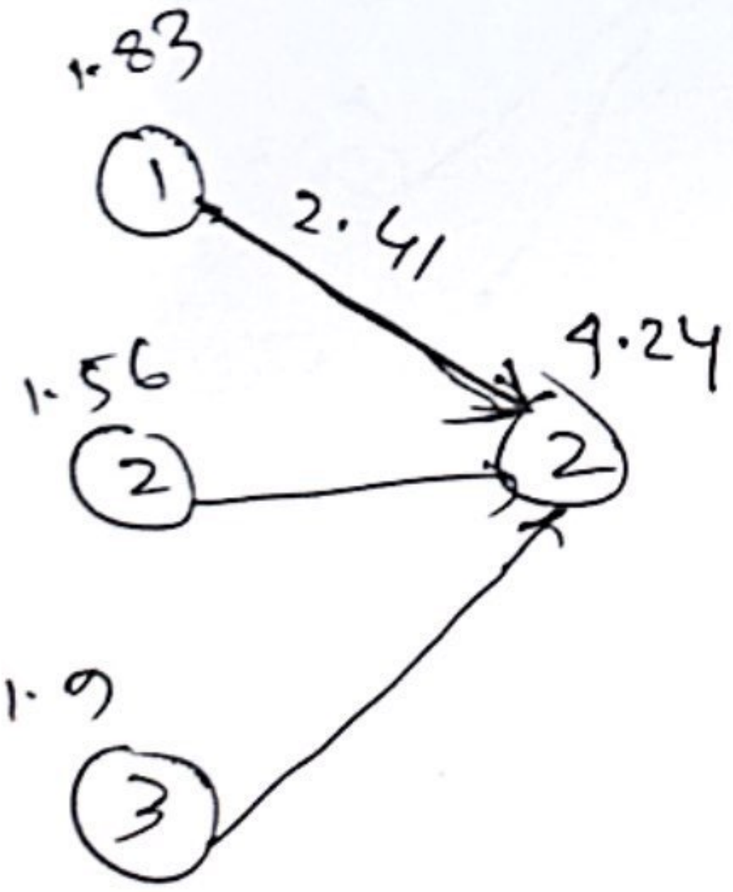
H	T	
1 → 1		$-\ln(0.9 \times 0.6) = 1.23$
2 → 1		$-\ln(0.5 \times 0.6) = 1.2$
3 → 1		$-\ln(0.9 \times 0.6) = 1.93$

As 2 → 1 is less sum

~~or probabil~~

So, 2 → 1 is selected

and sum = 1.56 + 1.2 = 2.76



H	T	
1 → 2		$-\ln(0.3 \times 0.3) = 2.41$

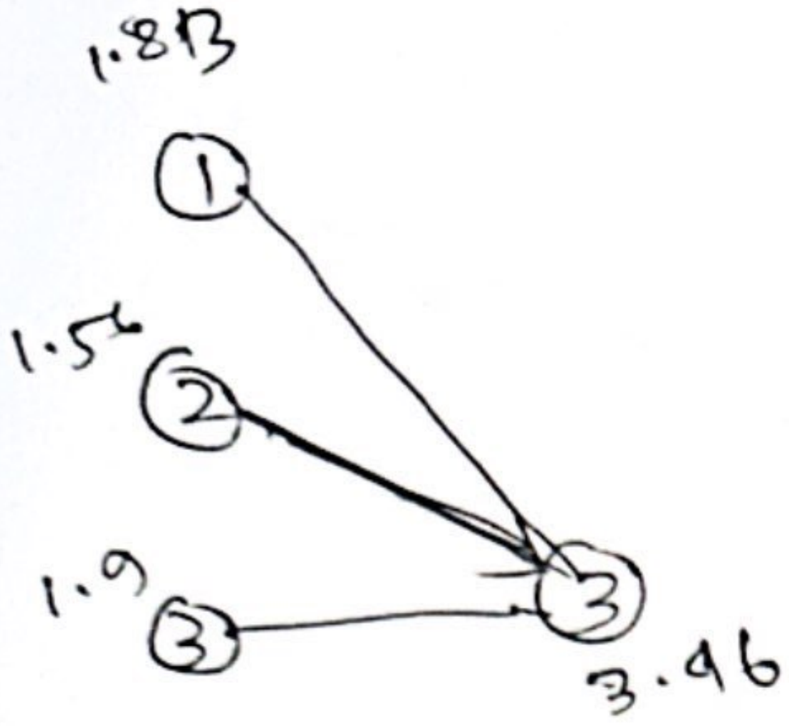
sum = 1.83 + 2.41 = 4.24

2 → 2 $-\ln(0.2 \times 0.3) = 2.81$

sum = 1.56 + 2.81 = 4.37

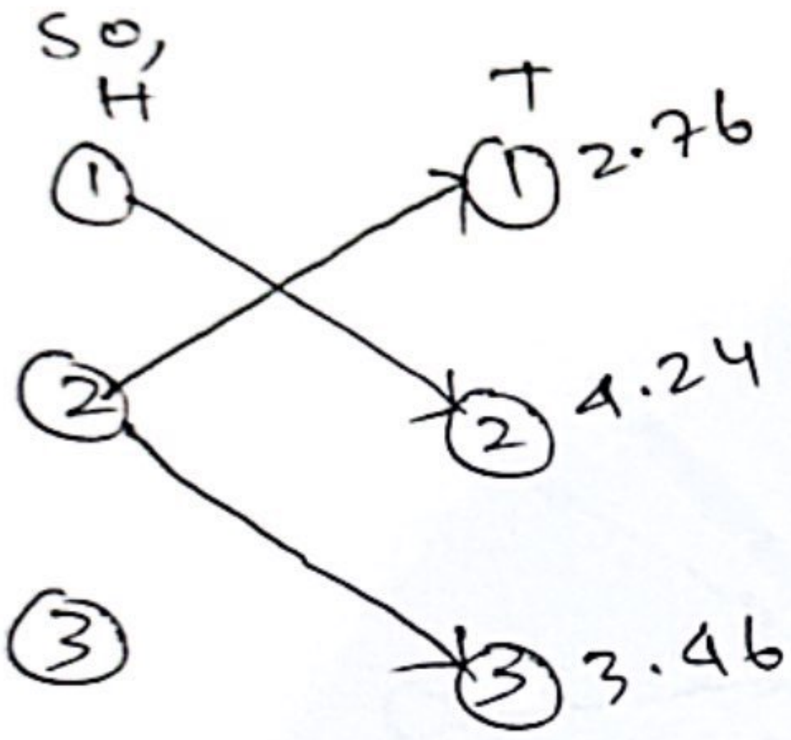
3 → 2 $-\ln(0.2 \times 0.3) = 2.81$

sum = 1.9 + 2.81 = 4.71

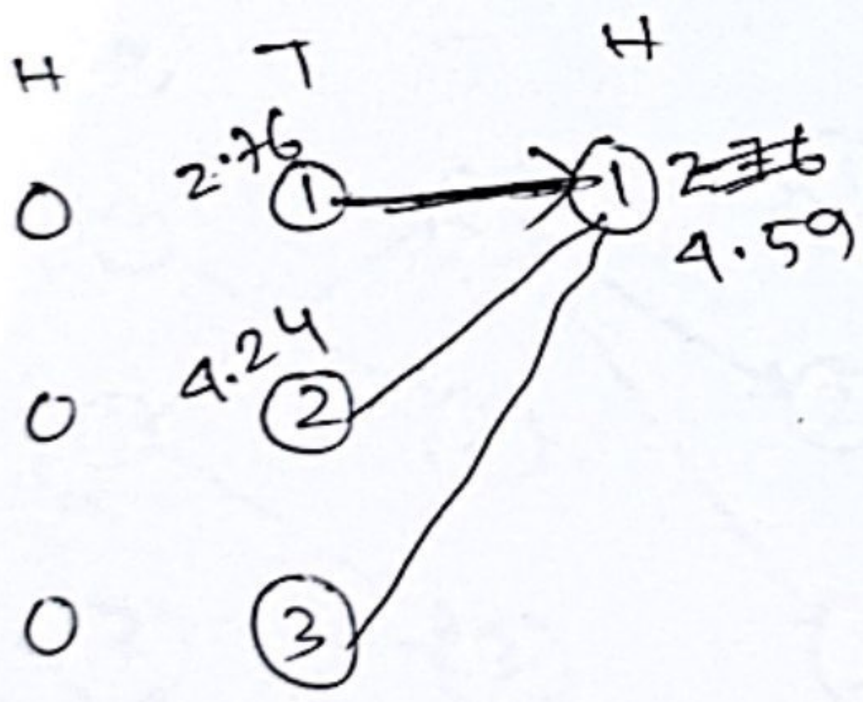


H	T	
1	3	$-\ln(0.3 \times 0.5) = 1.9$
⟹		sum $= 1.83 + 1.9 = 3.73$
2	3	$-\ln(0.3 \times 0.5) = 1.9$
		sum $= 1.56 + 1.9 = 3.46$
3	3	$-\ln(0.4 \times 0.5) = 1.61$
		sum $= 1.9 + 1.61 = 3.51$

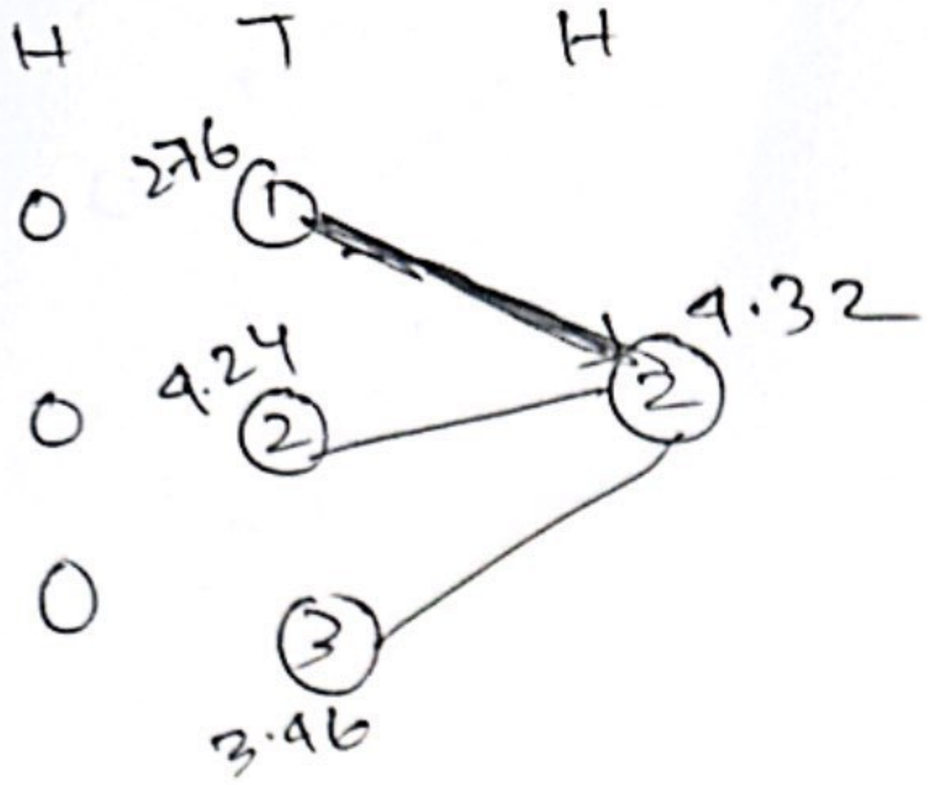
So,



For 3rd output : H



HT	H	
1	1	$-\ln(0.4 \times 0.4) = 1.83$
		sum $= 2.76 + 1.83 = 4.59$
HT	H	
2	1	$-\ln(0.5 \times 0.4) = 1.61$
		sum $= 4.24 + 1.61 = 5.85$
HT	H	
3	1	$-\ln(0.4 \times 0.4) = 1.83$
		sum $= 3.96 + 1.83 = 5.29$



H T H
1 → 2 $-\ln(0.3 \times 0.7) = 1.56$

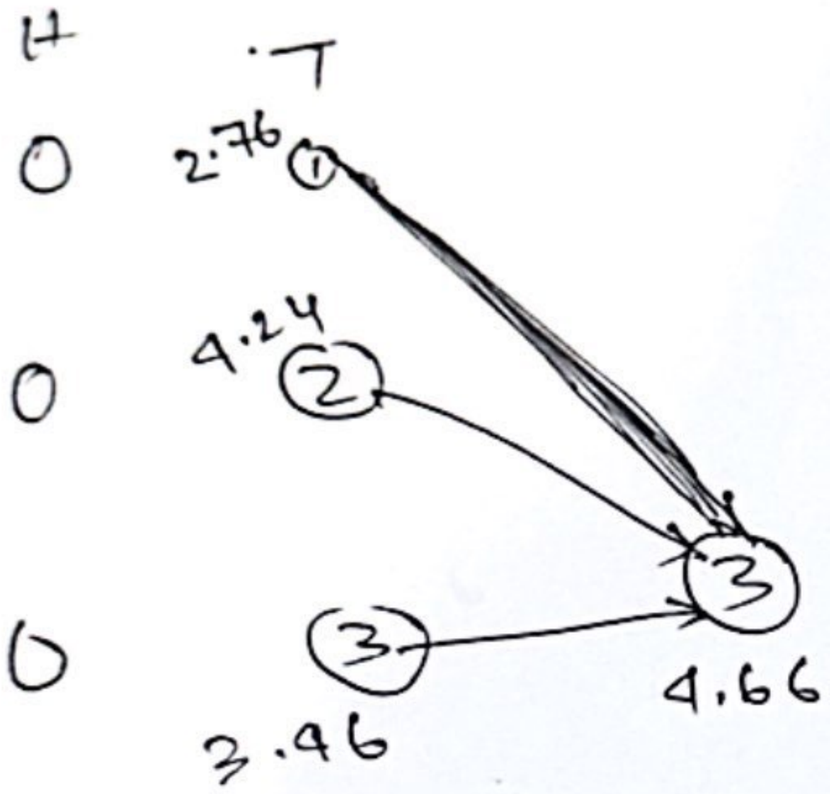
sum = 2.76 + 1.56 = 4.32

H T H
2 → 2 $-\ln(0.2 \times 0.7) = 1.97$

sum = 4.24 + 1.97 = 6.21

H T H
3 → 2 $-\ln(0.2 \times 0.7) = 1.97$

sum = 3.96 + 1.97 = 5.93



H T H
1 → 3 $-\ln(0.3 \times 0.5) = 1.9$

sum = 2.76 + 1.9 = 4.66

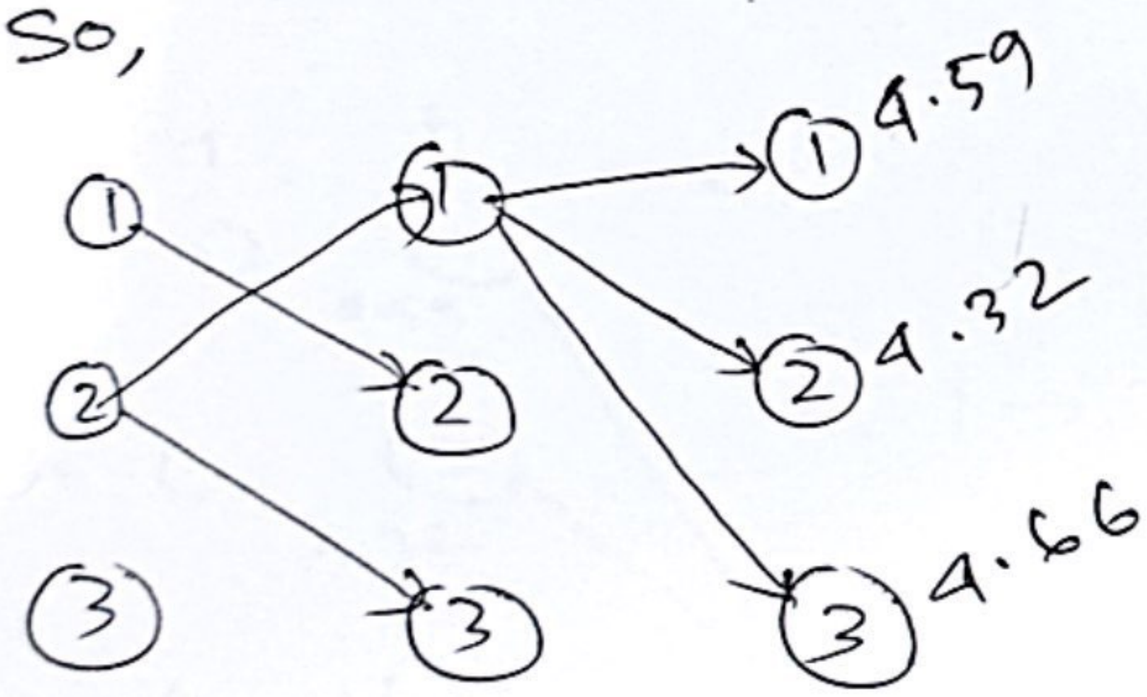
H T H
2 → 3 $-\ln(0.3 \times 0.5) = 1.9$

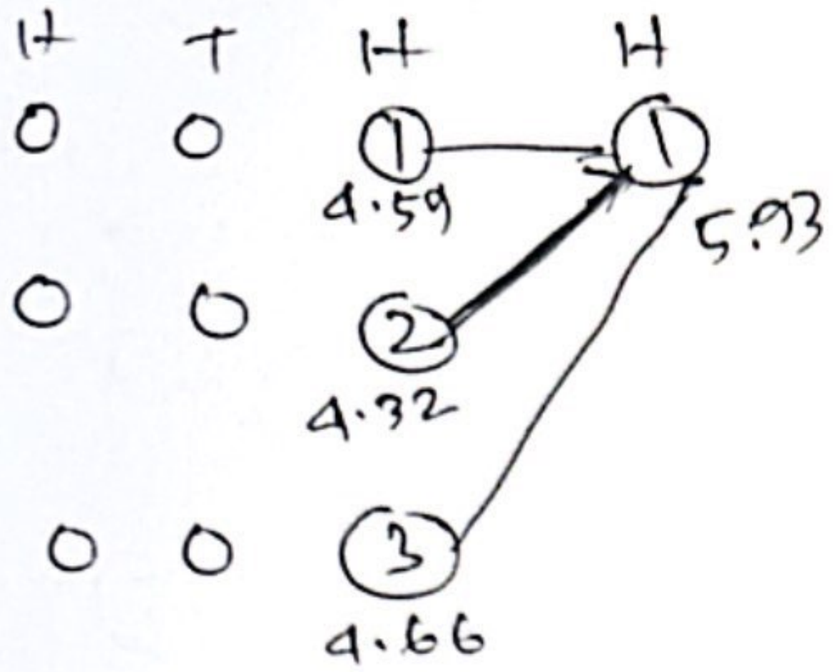
sum = 4.24 + 1.9 = 6.14

H T H
3 → 3 $-\ln(0.4 \times 0.5) = 1.61$

sum = 3.46 + 1.61 = 5.07

So,





HTH H
1 → 1 $-\ln(0.4 \times 0.4) = 1.83$

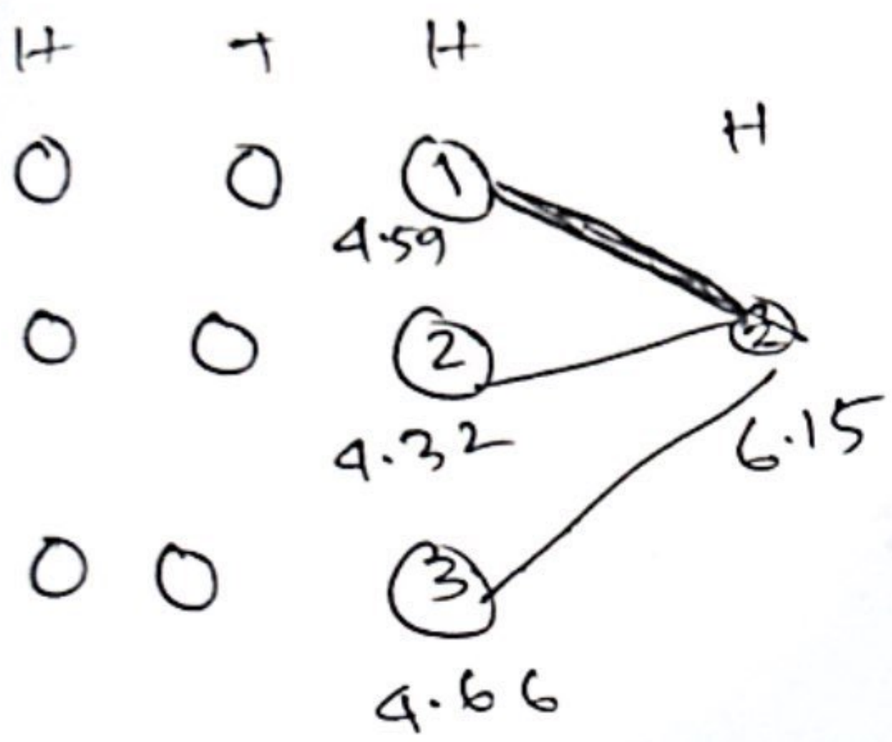
sum = 4.59 + 1.83 = 6.42

2 → 1 $-\ln(0.5 \times 0.4) = 1.61$

sum = 4.32 + 1.61 = 5.93

3 → 1 $-\ln(0.4 \times 0.4) = 1.83$

sum = 4.66 + 183 = 6.49



1 → H 2 $-\ln(0.3 \times 0.7) = 1.56$

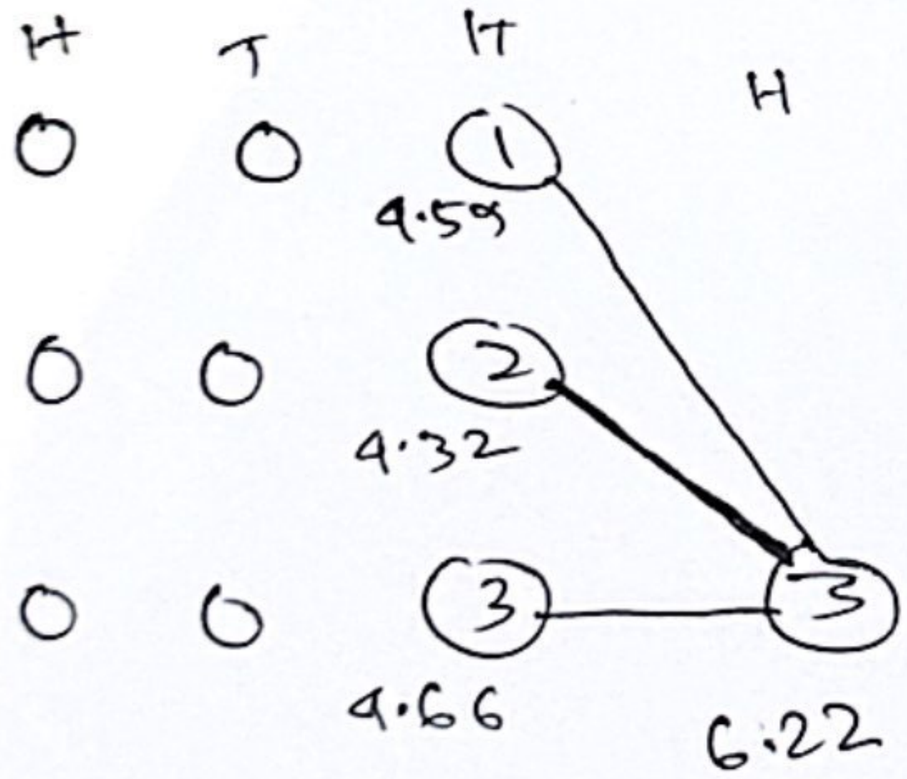
sum = 4.59 + 1.56 = 6.15

2 → H 2 $-\ln(0.2 \times 0.7) = 1.97$

sum = 4.32 + 1.97 = 6.29

3 → H 2 $-\ln(0.2 \times 0.7) = 1.97$

sum = 4.66 + 1.97 = 6.63



1 → H 3 $-\ln(0.3 \times 0.5) = 1.9$

sum = 4.59 + 1.9 = 6.49

2 → H 3 $-\ln(0.3 \times 0.5) = 1.9$

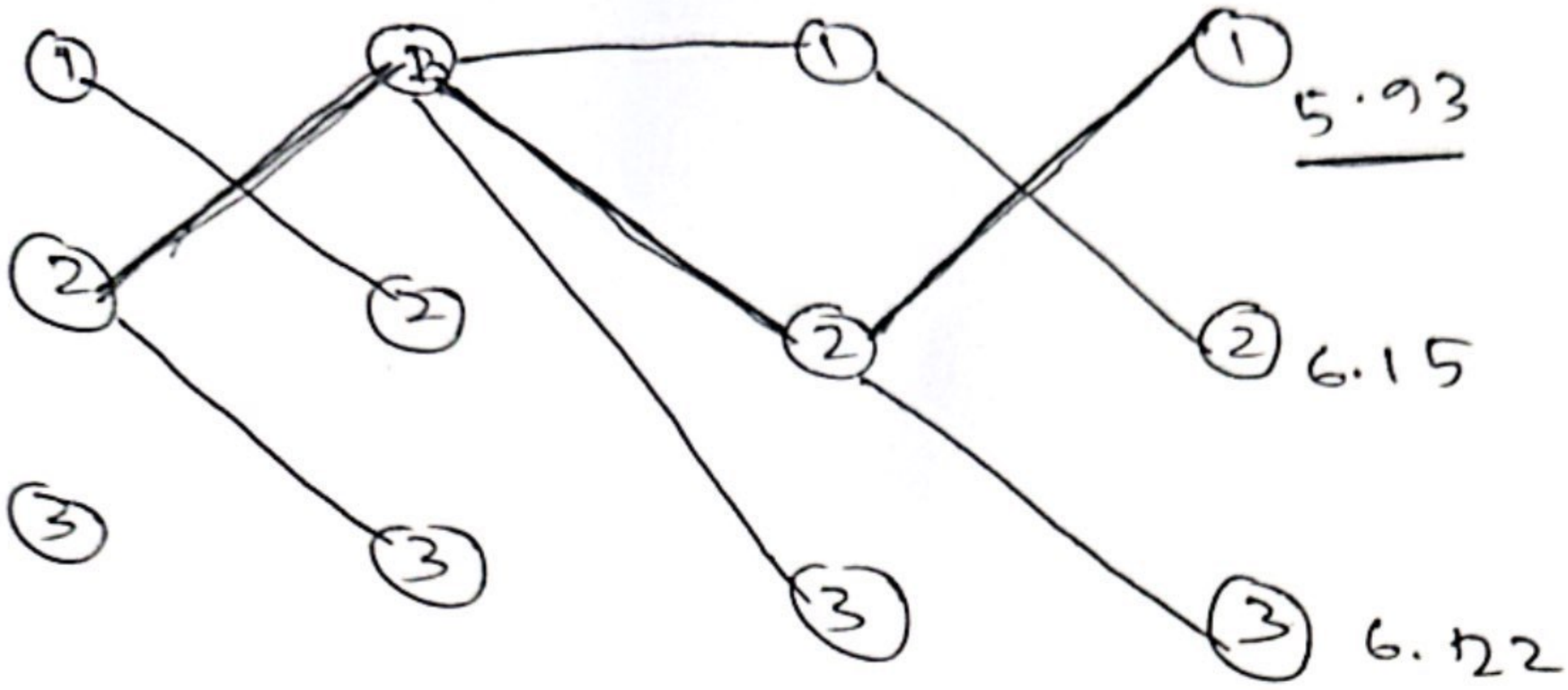
sum = 4.32 + 1.9 = 6.22

=

3 → H 3 $-\ln(0.9 \times 0.5) = 1.61$

sum = 4.66 + 1.61 = 6.27

So Final route



So the route or sequence is

$= 2 - 1 - 2 - 1$

So, probability $= e^{-5.93}$

$= 2.66 \times 10^{-3}$

Probability of ~~Modno~~ HTHH

$$\alpha_t(i) = P(O_1\ O_2\ O_3 \cdots O_t,\ i_t = i)$$

H

$$\alpha_1(i) = \pi_i b_i(O_1) \quad \therefore \alpha$$

$$\therefore \alpha_1(1) = 0.4 \times 0.4 = 0.16$$
$$(\pi_1) \times (b_i)$$

$$\alpha_1(2) = \pi_2 \times b_2(O_1) = 0.3 \times 0.7 = 0.21$$

$$\alpha_1(3) = \pi_3 \times b_3(O_1) = 0.3 \times 0.5 = 0.15$$

AW

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\ a_{ij}\right] b_j\ O_{(t+1)}$$

T

$$\therefore \alpha_2(1) = \left[\alpha_1(1) + a_{11} + \alpha_1(2) \times a_{21} + \alpha_1(3) \times a_{31}\right) \times b_i(O_2)$$

$$= [0.16 \times 0.4 + 0.21 \times 0.5 + 0.15 \times 0.4] \times 0.6$$

$$= 0.1374$$

$$\alpha_2(2) = [\alpha_1(1) \times a_{12} + \alpha_1(2) \times a_{22} + \alpha_1(3) \times a_{32}] b_2(O_2)$$

$$= (0.16 \times 0.3 + 0.21 \times 0.2 + 0.15 \times 0.2) \times 0.3$$

$$= 0.036$$

$$\alpha_2(3) = [\alpha_1(1) \times a_{13} + \alpha_1(2) \times a_{23} + \alpha_1(3)\ a_{33}] b_3(O_2)$$

$$= (0.16 \times 0.3 + 0.21 \times 0.3 + 0.15 \times 0.4) \times 0.5$$

$$= 0.0955$$

H

$$\alpha_3(1) = \left[\alpha_2(1)\times a_{11} + \alpha_2(2)\times a_{21} + \alpha_2(3)\times a_{31}\right] \times b_i(O_3)$$

$$= \left(0.1374\times 0.4 + 0.036\times 0.5 + 0.0855\times 0.9\right) \times 0.4$$

$$= 0.043$$

$$\alpha_3(2) = \left[\alpha_2(1)\times a_{12} + \alpha_2(2)\times a_{22} + \alpha_2(3)\times a_{32}\right] \times b_2(O_3)$$

$$= \left(0.1374 * 0.3 + 0.036\times 0.2 + 0.0855\times 0.2\right) \times 0.7$$

$$= 0.046$$

$$\alpha_3(3) = \left[\alpha_2(1)\times a_{13} + \alpha_2(2)\times a_{23} + \alpha_2(3)\times a_{33}\right] \times b_3(O_3)$$

$$= \left(0.1374\times 0.3 + 0.036\times 0.3 + 0.0855\times 0.9\right) \times 0.5$$

$$= 0.043$$

H

$$\alpha_4(1) = \left[\alpha_3(1)\times a_{11} + \alpha_3(2)\times a_{21} + \alpha_2(3)\times a_{31}\right] \times b_1(O_4)$$

$$= \left(0.043\times 0.4 + 0.046\times 0.5 + 0.043\times 0.9\right) \times 0.4$$

$$= 0.023$$

$$\alpha_4(2) = \left[\alpha_3(1)\times a_{12} + \alpha_3(2)\times a_{22} + \alpha_3(3)\times a_{32}\right] \times b_2(O_4)$$

$$= \left(0.043\times 0.3 + 0.046\times 0.2 + 0.043\times 0.2\right) \times 0.7$$

$$= 0.0215$$

$$\alpha_4(3) = \left(0.043\times 0.3 + 0.046\times 0.3 + 0.043\times 0.9\right) \times 0.5$$

$$= 0.022$$

$$P(HTHH) = \alpha_4(1) + \alpha_4(2) + \alpha_4(3)$$

$$= 0.023 + 0.0215 + 0.022$$

$$= 0.0665$$

8 / Previous CT-3 /

Phase - 1

Input size = 227 x 227 x 3

Number of filters = 100

filter size = [redacted] x 21 x 21 x 3

Stride = 4

Zero padding = 1

$$\text{output size} = \frac{W - F + 2P}{S} + 1$$

$$= \frac{227 - 21 + 2 \times 1}{4} + 1$$

$$= 53$$

∴ After 1st phase output of the convolution layer is = 53 x 53 x 100

For 2nd phase

Input size = 53 × 53 × 100

Number of filters = 50

filter size = 7 × 7 × 100

stride = 2

zero padding = 0

$$\text{output size} = \frac{53 - 7 + 2\times 0}{2} + 1$$

$$= 24$$

∴ convolution Layer = 24 × 24 × 50

For 3rd phase

INput size = 24 × 24 × 50

filters = 10

filter size = 5 × 5 × 50

stride = 1

zero padding = 0

$$\text{output size} = \frac{24 - 5 + 2\times 0}{1} + 1$$

$$= 20$$

convolution Layer = 20 × 20 × 10

Total number of neuron =

$(53 \times 53 \times 100) + (24 \times 24 \times 50) + (20 \times 20 \times 10)$

$= 313700$ (Ans)

9| Implimentation of Matrix Multiplication

of 1st phase

output : 53 × 53 × 100

filter size also : 21 × 21 × 3

So, First Matrix, (1323×2809)

$$A = \begin{bmatrix} (1,1) & (1,2) & (1,3) & \cdots & (1,2809) \\ (2,1) & & & & \\ \vdots & & & & \\ (1323,1) & & & & \end{bmatrix}$$

For 2nd Matrix (100×1323)

$$B = \begin{bmatrix} (1,1) & (1,2) & (1,3) & \cdots & (1,1323) \\ (2,1) & & & & \\ \vdots & & & & \\ (100,1) & & & & \end{bmatrix}$$

$$\therefore AXB = 100 \times 2809$$

$$\begin{bmatrix} (1,1) & (1,2) & \cdots & (1,2809) \\ (2,1) & & & \\ \vdots & & & \\ (100,1) & & & \end{bmatrix}$$

Q1 (2nd Half)

If we want to avoid pooling for downsizing the output of a convolutional layer, a common alternative is to use a convolutional layer with a larger stride

$$\text{output size} = \frac{w - F + 2P}{\text{Stride}} + 1$$

$$\therefore \text{output size} \propto \frac{1}{\text{stride}}$$

so, by increasing stride we can downsizing the output.

10/

Forward Method

$$\alpha_t(i) = P(O_1\ O_2\ O_3 \cdots O_T \mid \lambda)$$

for $\alpha_1(1) = \pi_1 b_1(O_1)$ [1 observed, State = 1

$\alpha_1(2) = \pi_2 b_2(O_1)$ [1, 2 observed State = 2

$\alpha_2(2) =$ ~~$\pi_2 b_2(O_2)$~~ [1, 2 observed State = 2

$\therefore \alpha_1(i) = \pi_i b_i(O_1)$ $\quad 1 \le i \le N$ $\longrightarrow (i)$

for $t = 1, 2, 3, 4$

$$\sum_{i=1}^{4} \alpha_T(i) = \alpha_4(1) + \alpha_4(2) + \alpha_4(3) + \alpha_4(4)$$

~~where~~ α.

$1 \le j \le N$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\ a_{ij}\right] b_j(O_{t+1})$$

$$\therefore P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

$$P(A,B) = P(A|B)\,P(B)$$

As

$$\alpha_2(j) = P(O_1, O_2, i_2 = j)$$

$$= \sum P(O_2, i_2 = j \mid O_1, \text{from state } i)$$

$$P(O_1 \text{ from state } i)$$

$$= \sum P(O_2 \mid i_2 = j, O_1 \text{ from state } i)$$

$$P(i_2 = j \mid O_1 \text{ from state } i)\; P(O_1 \mid i_1 = i)$$

$$P(i_1 = i)$$

$$= \sum \left[P(O_2 \mid i_2 = j) \left(P(i_2 = j \mid i_1 = i) \right) \right]$$

$$\left[P(O_1 \mid i_1 = i)\; P(i_1 = 1) \right]$$

$$= \sum b_j(O_2)\, a_{ij} \times b_i(O_1)\, \pi_i$$

$$= \left[\sum \pi_i\, b_i(O_1)\, a_{ij} \right] b_j(O_2)$$

$$= \left[\sum a_1(i)\, a_{ij} \right] b_j(O_2)$$

Complexity : $(N+1) \times (T-1) \times N + N$

$\approx N^2 T$ if $T = 100$, $N = 5$

$\simeq 3000$

Hidden Markov Model

Calculate the probability of a sequence of observation :

$$P(O, O_2\ O_3\ O_4 \dots O_T) = ?$$

means $P(O, O_2\ O_3 \dots O_T \mid \lambda) = ?$

$$\Rightarrow P(O \mid \lambda) = ?$$

Ans: Let, $I = I_1, I_2, I_3 \dots I_T$

we know,

$$P(O \mid I, \lambda) = bi_1(O_1) \times bi_2(O_2) \times \dots \times bi_T(O_T)$$

$$P(I \mid \lambda) = \pi_{i_1} a_{i_1 i_2} a_{i_2 i_3} a_{i_3 i_4} \dots a_{i_{T-1} i_T}$$

$$\therefore P(O \mid \lambda) = \sum P(O \mid I, \lambda)\ P(I \mid \lambda)$$

$$= \sum \pi_{i_1} b_{i_1}(O_1) a_{i_1 i_2} b_{i_2}(O_2) a_{i_2 i_3} b_{i_3}(O_3) \dots a_{i_{T-1} i_T} b_{i_T}(O_T)$$

Complexity :

$$P(O \mid I, \lambda) = T$$

$$P(I \mid \lambda) = T$$

Total states = N

$(T+T) \times N^T$ ——— [$N^1 \times N^2 \times N^3 \times N^4 \ldots$]

$= 2T N^T$

∴ if $T=100$, $N=5 \approx 10^{75}$

Forward method is better for Markov Model